

HW3 – ARM 명령어

1) Signed 연산을 수행했다고 가정한다.

$r2 = 0x76543210$, $r3 = 0x98765432$

ADDS명령어:

$r1 = r2 + r3 = 0x0ECA8642$ (16진수)

N = 0 유지 (연산의 결과가 양수이다.)

Z = 0 유지 (연산의 결과가 0이 아님)

C = 1 유지 (32비트를 넘은 연산)

V = 0 유지 (양수와 음수의 덧셈)

SUBS명령어:

$r1 = r2 - r3 = r2 + r3(2의보수) = 0x76543210 + 0x6789ABCE$

$= 0xDDDDDDDE$

N = 1 바뀜 (연산의 결과가 음수)

Z = 0 유지 (연산의 결과가 0이 아님)

C = 0 바뀜 (32비트를 넘는 연산이 아님)

V = 1 바뀜 (양수와 양수의 덧셈이 음수로 바뀜)

2) 데이터 전송 명령어

LDR r1, [r2,#20]: r1에 0x17161514을 저장

STR r1, [r2,#20]: 메모리[2020]에 0xE8, [2021]에 0x03, [2022]에 0x00, [2023]에 0x00을 저장

LDRH r1, [r2,#20]: r1에 0x00001514를 저장

LDRHS r1, [r2,#20]: r1에 0x00001514를 저장(sign bit 확장)

STRH r1, [r2,#20]: 메모리[2020]에 0xE8, [2021]에 0x03을 저장

LDRB r1, [r2,#20]: r1에 0x00000014를 저장

LDRBS r1, [r2,#20]: r1에 0x00000014를 저장(sign bit 확장)

STRB r1, [r2,#20]: 메모리[2020]에 0xE8을 저장

SWP r1, [r2,#20]: r1에 0x17161514을 저장, 메모리[2020]에 0xE8, [2021]에 0x03, [2022]에 0x00, [2023]에 0x00을 저장

MOV r1, r2: r1에 2000을 저장

3) 논리 명령어

ANDS r1,r2,r3:

$r1 = r2 \& r3 = 0x10541010$

N = 0 유지

Z = 0 유지

C = 0 바뀜

ORRS r1,r2,r3:

$r1 = r2 \mid r3 = 0xFE767632$

N = 1 바뀜

Z = 0 유지

C = 0 바뀜

MVN r1,r2:

$r1 = \sim r2 = 0x89ABCDEF$

LSL r1,r2,#10:

$r1 = r2 \ll 10 = 0x50C84000$

LSR r1,r2,#10:

$r1 = r2 \gg 10 = 0x001D950C$

4) 조건부 분기

① **CMP r1, r2:**

연산은 SUB명령어와 같다. 플래그만 변경한다.

N = 1 바뀜 (연산의 결과가 음수)

Z = 0 유지 (연산의 결과가 0이 아님)

C = 0 바뀜 (32비트를 넘는 연산이 아님)

V = 1 바뀜 (부호가 바뀌는 오버플로우 발생)

Z플래그의 값이 0이므로 EQ조건에 맞지 않는다.

따라서 PC값은 1008로 변경된다.

② **CMP r1, r2:**

연산은 SUB명령어와 같다. 플래그만 변경한다.

N = 0 유지 (연산의 결과가 음수가 아님)

Z = 1 바뀜 (연산의 결과가 0이다)

C = 0 바뀜 (32비트를 넘는 연산이 아님)

V = 0 유지 (오버플로우 발생하지 않음)

Z플래그의 값이 1이므로 EQ조건에 맞는다.

따라서 PC값은 $1004 + 8 + 100 = 1112$ 로 변경된다.

5) 무조건 분기

① PC값은 $1000 + 8 + 200 \times 4 = 1808$ 로 변경된다.

② PC값은 $1000 + 8 + 200 \times 4 = 1808$ 로 변경된다.

LR에는 1004가 저장된다.